DETAILED ACTION

1. This action is responsive to the application filed 10/29/1996 and the interview on 09/04/2012.

Claims 1-4, 6-8, and 11-14 are pending in the application. Claims 1-4, 6-8, and 11-14 have been examined and allowed.

Information Disclosure Statement

2. The Applicants' Information Disclosure Statements, filed 06/29/1994 and 10/29/1996, have been received, entered into the record, and considered. Copies of PTO 1449 forms are attached.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. William A. Kinnaman (Reg. No. 27,650) on 09/04/2012.

In the Specification:

The Abstract has been amended as follows:

~~A method of coordinating the quiescing of the additional threads of a multithreaded application to facilitate the handling of an external event by a first thread without interference from the additional threads~~. In response to the detection of an external event ~~by the first thread~~, the first thread sends a quiesce event to each additional thread of the application and suspends execution. The quiesce event may be either a suspension event requesting suspension of the additional threads or a termination event requesting termination of the additional threads. Each additional thread, upon receiving the quiesce event, checks its environment to determine whether it is holding any critical system resource. If the additional thread determines that is not holding any critical system resource and that it is therefore safe to quiesce, the additional thread quiesces. Before quiescing, the last additional thread to quiesce resumes the first thread, which is now free to perform critical operations without interference from the additional threads. If the quiesce type is suspension, the first thread resumes the additional threads upon completing its critical operations, whereupon the application resumes its normal operation.

In the Claims:

This listing of claims will replace all prior versions and listings of claims in the application.

1. (Currently Amended) In a computer system in which a first thread and a second thread of a user application execute concurrently in a common address space, a method of processing an application event in response to the detection of said application event by said first thread, comprising the steps of:

said first thread, in response to detecting said application event:

sending a quiesce event to said second thread to cause said second thread to quiesce; and

suspending execution until said second thread has quiesced in response to the quiesce event sent to [[that]] <u>said second</u> thread; and

said second thread, in response to receiving said quiesce event:

determining whether it is holding any resource required by another thread;

quiescing only if it determines that it is not holding any resource required by said another thread; and

upon quiescing, resuming execution of said first thread to process said application event.

2. (Original) The method of Claim 1 wherein said second thread is one of a plurality of additional threads executing concurrently with said first thread in said address space, said quiesce event being sent from said first thread to each of said additional threads.

3. (Currently Amended) The method of Claim 2 wherein execution of said first thread is suspended until each of said additional threads has quiesced in response to the quiesce event sent to ~~that thread~~ said each of said additional threads.

4. (Original) The method of Claim 3 wherein the last of said additional threads to quiesce resumes execution of said first thread.

5. (Cancelled)

6. (Previously Presented) The method of Claim 1 wherein said quiesce event is a termination event causing said second thread to terminate execution.

7. (Previously Presented) The method of Claim 1 wherein said quiesce event is a suspension event causing said second thread to suspend execution.

8. (Previously Presented) The method of Claim 1 wherein said step of sending a quiesce event from said first thread to said second thread comprises the step of interrupting the execution of said second thread to give control to a quiesce exit routine.

9. -10. (Cancelled)

11. (Currently Amended) The method of Claim 1 wherein said second thread releases any resource required by said another thread that is held by said second thread before quiescing said second thread.

12. (Currently Amended) In a computer system in which a first thread and a second thread of a user application execute concurrently in a common address space, a method of processing an application event in response to the detection of said application event by said first thread, comprising the steps of:

said first thread, in response to detecting said application event:

sending a suspension event to said second thread to cause said second thread to suspend; and

suspending execution until said second thread has suspended in response to the suspension event sent to [[that]] said second thread;

said second thread, in response to receiving said quiesce event:

determining whether it is holding any resource required by another thread;

quiescing only if it determines that it is not holding any resource required by said another thread; and

upon quiesing resuming execution of said first thread to process said application event; and

said first thread resuming said second thread following the processing of said application event by said first thread.

13. (Previously Presented) The method of Claim 12 wherein a plurality of additional threads execute concurrently with said first thread in said address space, said suspension event being sent from said first thread to each of said additional threads.

14. (Currently Amended) The method of Claim 13 wherein execution of said first thread is suspended until each of said additional threads has suspended in response to the suspension event sent to ~~that thread~~ said each of said additional threads.

REASONS FOR ALLOWANCE

4. Claims 1-4, 6-8, and 11-14 are allowed.

The following is an examiner's statement of reasons for allowance:

Interpreting the claims in light of the specification, Examiner finds the claimed invention is patentably distinct from the prior art of record.

The prior art does not expressly teach or render obvious the invention as recited in independent Claims 1 and 12.

The features as recited in independent claim 1: *"said second thread, in response to receiving said quiesce event: determining whether it is holding any resource required by another thread; quiescing only if it determines that it is not holding any resource required by said another thread; and upon quiescing, resuming execution of said first*

thread to process said application event", **when taken in the context of the claims as a whole, were not uncovered in the prior art teachings.**

The features as recited in independent claim 12: *"said second thread, in response to receiving said quiesce event: determining whether it is holding any resource required by another thread; quiescing only if it determines that it is not holding any resource required by said another thread; and upon quiesing resuming execution of said first thread to process said application event; and said first thread resuming said second thread following the processing of said application event by said first thread",* **when taken in the context of the claims as a whole, were not uncovered in the prior art teachings.**

Dependent claims 2-4, 6-8, 11, 13, and 14 are allowed as they depend upon allowable independent claims.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Contact Information

Any inquiry or a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: (571) 272-2100.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H. NGUYEN whose telephone number is (571) 272-3765. The examiner can normally be reached on Monday- Friday from 9:00AM-5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, LEWIS BULLOCK can be reached at (571) 272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VAN H NGUYEN/
Primary Examiner, Art Unit 2199